МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

КАФЕДРА ИИТ

ОТЧЁТ

по лабораторной работе №3

**«Механизм обработки исключений в С++»**

Выполнил:

студент 2 курса

группы ПО-9

Мисиюк Алексей Сергеевич

Проверил:

Козик И. Д.

labrab3.cpp

#include <iostream>

#include "CanThrowError.h"

#include "TestError.h"

#include "RandomTestError.h"

void task4();

void task9();

/\* Variant #15

\* 15) задания 4, 9

\*/

int main()

{

task4();

task9();

return 0;

}

// 4) Напишите программу на С++, которая показывает, что все деструкторы объектов,

// созданных в блоке, вызываются прежде, чем в этом блоке генерируется исключение.

void task4() {

try {

std::cout << "Creating obj1 and obj2\n";

CanThrowError obj1, obj2;

std::cout << "Throwing error\n";

obj1.throwTestError();

CanThrowError obj3;

}

catch (TestError) {

std::cout << "Task4's catch called.\n";

}

}

// 9) Напишите программу на С++, которая использует многоуровневую иерархию наследования

// классов исключения для создания ситуации, в которой важна последовательность обработчиков исключений.

void task9() {

for (int i = 0; i < 5; i++) {

try {

if (rand() % 2) {

std::cout << "Random case: ";

throw RandomTestError();

}

std::cout << "Normal case: ";

throw TestError();

}

//catch (RandomTestError) {

// std::cout << "Task 9's RandomTestError catch called.\n";

//}

catch (TestError) {

std::cout << "Task 9's TestError catch called.\n";

}

}

}

CanThrowError.h

#pragma once

#include <iostream>

#include "TestError.h"

// Has a throwTestError method.

// Also, when destructor called, log it in cout.

class CanThrowError

{

public:

void throwTestError();

~CanThrowError();

};

CanThrowError.cpp

#include "CanThrowError.h"

void CanThrowError::throwTestError() {

throw TestError();

}

CanThrowError::~CanThrowError() {

std::cout << "Destructor called.\n";

}

TestError.h

#pragma once

class TestError

{

};

RandomTestError.h

#pragma once

#include "TestError.h"

class RandomTestError : public TestError

{

};

Результат:

Creating obj1 and obj2

Throwing error

Destructor called.

Destructor called.

Task4's catch called.

Random case: Task 9's TestError catch called.

Random case: Task 9's TestError catch called.

Normal case: Task 9's TestError catch called.

Normal case: Task 9's TestError catch called.

Random case: Task 9's TestError catch called.